

Git och Github

Innehåll

1. Vad är Git och Github?
2. Vad är skillnaden mellan Git och Github?
3. Koncept i Git
 - a. Repository
 - b. Commit
 - c. Push & Pull
4. Kort om avancerade koncept
 - a. Branch
 - b. Merge
 - c. Rebase
5. Demonstration
 - a. Terminal
 - b. Visual Studio Code
 - c. IntelliJ

Vad är Git?

- Versionshantering system
 - Ett sätt att hålla koll på filer och historik
 - Ett sätt att samarbeta och jobba på samma projekt som andra
 - Ett sätt att spara backups av projekt och filer
- Lokalt program som installeras på datorn
- Sätt att arbeta med Git
 - Terminal
 - IntelliJ
 - Visual Studio Code



Vad är GitHub?

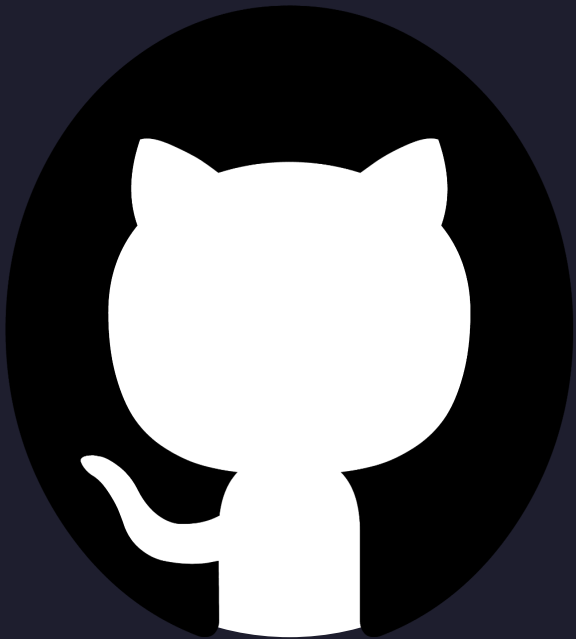
- Plattform och webb-baserad hosting tjänst för Git
- Använder Git som system för att hantera filer
- Används för att ladda upp och spara filer och projekt online
- Projekt kan delas mellan flera personer för att tillåta samarbete
- Har lite extra funktionalitet
 - Pull requests
 - Issues



Andra plattformar

- GitHub (mest populär)
- GitLab
- BitBucket

Dessa fungerar på nästan exakt samma sätt. Plattformen är valfri.



Skillnad mellan Git och GitHub

- Som skillnaden mellan fotografering och Instagram
 - Fotografering är handlingen av att ta bilder
 - Instagram är ett ställe man kan ladda upp och dela bilder
- Git fungerar utan GitHub, men GitHub fungerar inte utan Git

Installera och kom igång med Git

- Git måste först installeras ifrån <https://git-scm.com/downloads>
- Git måste sedan konfigureras med `git config`

I terminal/konsol:

```
git config --global user.name "My name"
```

```
git config --global user.email "my.name@email.com"
```

```
git config --global init.defaultBranch main
```

Koncept och termer

1. Repository (repo)
 - a. Local
 - b. Remote
2. Add & Commit
3. Push
4. .gitignore
5. Pull
6. Clone

Repository

- Förkortas ibland till 'repo'
- Samling med filer som bildar ett projekt
- Local och remote variant
- Innehåller typiskt sett kod filer men kan ha annat

Local

- Lokal version av ett projekt
- Skapas lokalt i terminal/konsol med: `git init`

Remote

- Online version (på GitHub) av ett projekt
- Måste först skapas på plattform (e.g. GitHub)

Local & remote repositories

- Git vet inte automatiskt om GitHub
- Lokala repos kopplas inte automatiskt till remote repos

Koppla ett lokalt repo till ett remote repo med: `git remote add origin <url till repo>`

Staging & commits

Staging/add

- Registrera ändringar i form av nya filer, ny kod eller raderingar
- Registrera/uppdatera en specifik fil i terminal: `git add filename.ext`
- Registrera/uppdatera alla filer i repo: `git add .`

Commit

- Sammanställer alla ändringar från 'add' till ett slags paket
- I terminal: `git commit -m "Description of my changes"`

Pushing

- Ladda upp alla ändringar (commits) till GitHub
- Kan endast göras om det finns en eller flera commits redo
- Kan endast göras om det inte finns några ändringar som måste hämtas först

Gör en push i terminal med: `git push -u origin main`

.gitignore

- Ibland så vill man inte ladda upp vissa filer, men de måste finnas där lokalt
 - Det är vanligt att ha “build” filer och lösenord lokalt, men dessa vill man inte ladda upp online
- Specificera vilka filer som skall “ignoreras” med en .gitignore
- Laddas upp med en vanlig commit

Exempel .gitignore:

private-image.png

my-passwords.txt

build/

Pulling

- Ladda ned alla ändringar (commits) från GitHub
- Kan endast göras om det finns en eller flera commits som man inte har lokalt
- Kan endast göras om det inte finns några ändringar lokalt
 - En commit måste skapas innan man kan hämta ändringar
 - Annars så skulle Git radera dem

Gör en pull i terminal med: `git pull origin main`

Kloning

- Ladda ned och kopiera ett remote repo från GitHub
- Om man klonar så behöver man inte skapa ett lokalt repository
 - Det görs automatiskt med git clone

Klona i terminal med: `git clone <url till repo>`

Demonstration

Terminal/konsol

Demonstration

Visual Studio Code

Demonstration

Intellij

Avancerade koncept och termer

1. Konflikter
2. Branch
3. Merge
4. Pull requests (GitHub)

Konflikter

- Uppstår när två commits ändrar på samma ställe
- Måste lösas manuellt och sedan laddas upp i en ny commit

Branches

- En separat och isolerad version av ett repository
- Alla repos har en 'main' branch
- Används för att bygga på med ny kod utan att förstöra 'main'
- Två branches kan slås ihop genom 'merging'

Skapa ny branch och byt till den i terminal med: `git checkout -b my-branch`

Byt till annan branch i terminal med: `git checkout main`

Merging

- Slår ihop två branches
- Git hanterar automatiskt processen men det kan uppstå konflikter
- Görs när man är klar med en branch och vill få in den i 'main'

Skapa ny branch och byt till den i terminal med: `git checkout -b my-branch`

Byt till annan branch i terminal med: `git checkout main`

Pull requests

- En form av merge som görs på GitHub
- Tillåter reviewing och konflikthantering
- Kan involvera flera personer och kommentarer

Demonstration

Terminal/konsol

Demonstration

Visual Studio Code

Demonstration

Intellij